UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

DATATERN, INC. Plaintiff, v. BLAZENT, INC., MICROSTRATEGY INC., CARL WARREN AND COMPANY INCORPORATED, LANCET SOFTWARE DEVELOPMENT, INC., AIRLINES REPORTING CORP., MAGIC SOFTWARE ENTERPRISES LTD., MAGIC SOFTWARE ENTERPRISES, INC., TERADATA CORPORATION, INFORMATICA CORPORATION, EPICOR SOFTWARE CORPORATION, and PREMIER, INC. Defendants.	1:11-cv-11970-FDS (Consolidated)

MICROSTRATEGY'S MEMORANDUM IN SUPPORT OF ITS MOTION FOR SUMMARY JUDGMENT OF INVALIDITY FOR UNPATENTABLE SUBJECT MATTER

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

I. INTRODUCTION

The claims of U.S. Patent No. 6,101,502 (the "'502 patent") recite a method for describing relationships between a database structure (a "schema") and a set of relationships among objects (an "object model"). These are processes that can be performed purely in one's head or with pen and paper. The claims then add well-known steps tying these mental processes to a particular method of accessing a computer database. But such conventional post-solution activity is insufficient to transform the claims into patent eligible subject matter. Under Supreme Court and Federal Circuit precedent, therefore, the claims are invalid under 35 U.S.C. § 101.

In May 2012, when this case was pending before Judge Stearns, MicroStrategy moved for judgment on the pleadings that the '502 patent is invalid for failure to recite patent-eligible subject matter.[1] The Court denied that motion without prejudice based on then-recent caselaw suggesting that it would often be preferable to complete claim construction before resolving patentability disputes arising under § 101, identifying two terms that might require construction ("interface objects" and "runtime engine"). Since that decision, several developments have rendered the threshold question of subject-matter eligibility ripe for resolution.

First, in June 2014, the Supreme Court held four patents invalid for claiming unpatentable subject matter. *Alice Corp. v. CLS Bank Int'l*, 134 S. Ct. 2347 (2014). Notably, during the parties' previous briefing on this issue (which pre-dated the Supreme Court opinion), DataTern explicitly likened its patented technology to that at issue in *Alice*. *Alice* also confirmed and clarified a two-part test for patentability, a test which the '502 patent cannot pass. Second,

[1] MicroStrategy's motion was filed shortly after the March 2012 scheduling conference where Judge Stearns *sua sponte* expressed doubt about patent validity under Section 101, citing *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366 (Fed. Cir. 2011) and *Mayo Collaborative Services v. Prometheus Labs, Inc.*, 132 S. Ct. 1289 (2012).

since *Alice*, district courts have routinely decided § 101 issues early in the case, without need for a separate claim construction proceeding.[2] And third, the '502 patent terms the Court identified in its decision to defer ruling on the patentability of the asserted claims have since undergone claim construction in the Southern District of New York, and Federal Circuit review.

In view of these developments, it is appropriate to resolve this issue now, before the parties undertake the significant burden and expense of discovery. The Court should therefore find each claim of the '502 patent invalid under § 101 and grant summary judgment of invalidity.

II. STATEMENT OF FACTS[3]

A. MicroStrategy's Previous Motion for Judgment Under § 101 and Subsequent Claim Construction Decisions.

On May 10, 2012, MicroStrategy moved for judgment on the pleadings requesting this Court declare the '502 patent invalid under § 101. (SOF ¶ 1.) DataTern responded by asserting the motion was premature because the Court had not construed the claims and that the '502 claims were patent eligible because, *inter alia*, "the claimed methods and systems of the '502 patent require computer systems and databases." (SOF ¶ 2.) Following MicroStrategy's reply brief, DataTern filed a Notice of Supplemental Authority bringing to the Court's attention a since-vacated Federal Circuit panel decision in *CLS Bank Int'l v. Alice Corp.*, wherein the Federal Circuit reversed a district court decision holding the patents invalid under § 101. 685

[2] By way of example, in the wake of *Alice,* courts promptly began addressing § 101 issues in motions for judgment on the pleadings. *See e.g., buySAFE, Inc. v. Google, Inc.*, 765 F.3d 1350 (Fed. Cir. 2014); *Loyalty Conversion Systems Corp. v. American Airlines, Inc.*, No. 2:13-cv-00655-WCB, 2014 WL 4364848 (E.D. Tex. Sept. 3, 2014); *Tuxis Technologies, LLC v. Amazon.com, Inc.*, No. 1:13-cv-01771-RGA, 2014 WL 4382446 (D. Del. Sept. 3, 2014); *Genetic Technologies Ltd. v. Laboratory Corp. of Am. Holdings*, No. 1:12-cv-01736-LPS-CJB, 2014 WL 4379587 (D. Del. Sept. 3, 2014).

[3] Further background facts and procedural history are found in MicroStrategy's co-pending motion for summary judgment of non-infringement. This motion focuses on background most relevant to the § 101 inquiry.

F.3d 1341 (Fed. Cir. 2012); (SOF ¶ 3.) DataTern argued that "[j]ust as the creation of a 'shadow credit record' or a 'shadow debit record' was sufficient to confer § 101 eligibility on the CLS Bank patents, the creation of 'interface objects' and their invocation by a 'runtime engine' are at least enough to pass the § 101 eligibility threshold in this case."[4] (SOF ¶ 4.)

On July 31, 2013, the Court denied without prejudice MicroStrategy's § 101 motion. (SOF ¶ 5.) The Court cited to then-recent caselaw for the proposition that it may be desirable to resolve claim construction disputes prior to addressing patent subject matter eligibility. (SOF ¶ 6.) The Court recognized that "the patentability issue in this case turns on the significance of certain claim elements, including interface objects and runtime engine, the court believes that it is prudent to follow the Federal Circuit's advice and construe the claims first." (SOF ¶ 7.)

Shortly after the Court's ruling in July 2013, the Southern District of New York construed the claims of DataTern's '502 patent in a co-pending case, including the terms "runtime engine" and "to create at least one interface object." (SOF ¶ 8.) Those constructions underpinned a judgment of non-infringement that DataTern appealed to the Federal Circuit. (SOF ¶ 9.) The Federal Circuit affirmed the New York judgment on grounds that didn't require it to reach the "runtime engine" construction, which was construed by the court to mean: "software that (i) the object oriented software application depends on to run, (ii) must be running to execute the object oriented software application, (iii) uses the map in its processing, and (iv) is not part of the object oriented software application." (SOF ¶ 10.) The New York court rejected DataTern's proposed construction of runtime engine, which was "software, which is not directly part of the

[4] The Federal Circuit subsequently granted a petition for rehearing *en banc* and ultimately reversed the panel decision, thus finding the patents-at-issue invalid under § 101. The Supreme Court granted *certiorari* and, in 2014, affirmed the finding that the *Alice* patents were invalid because they were directed to patent-ineligible subject matter.

object-oriented application, that the object-oriented application uses to access the relational database." (SOF ¶ 11.) Separately, on the later appeal of the judgment of non-infringement in the instant case, the Federal Circuit construed "to create at least one interface object" to mean "to instantiate at least one interface object from a class." (SOF ¶ 12.) The Federal Circuit did not separately construe the "interface object" portion of the claim. In New York, however, DataTern proposed that "interface object," standing alone, means "an object by which the object oriented application accesses the relational database via a runtime engine.'" (SOF ¶ 13.)

As shown below, this background resolves any questions that might be a predicate to reaching the subject-matter eligibility of the claims of the patent-in-suit. Even under DataTern's preferred constructions, the claims are directed to ineligible subject matter and are thus invalid.

B. The '502 Patent

1. Introduction to the Technology of the '502 Patent[5]

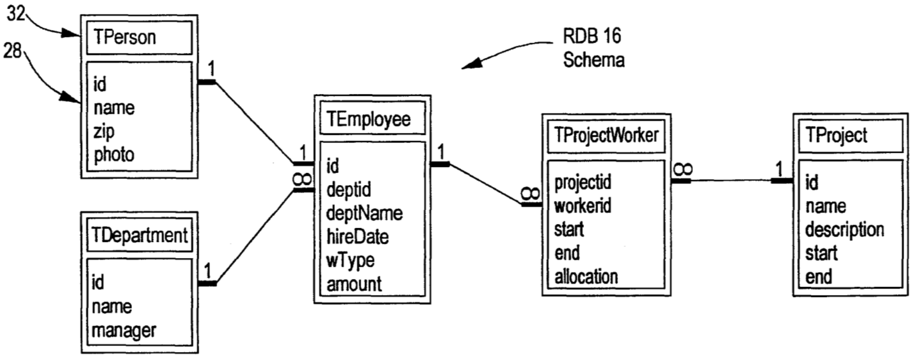
(a) Relational Databases

A database is a collection of stored information, for example, email messages. Data in a database is organized into "tables," including rows and columns. The rows specify particular data items (e.g., each specific email), and the columns specify information about those items (sender, recipient, subject, etc.). Most modern databases are "relational," meaning that tables in the database are related to one another. For example, email programs often keep contact information. Email messages and contacts are stored in separate tables. However, the sender column in the email table might relate to an email address column in the contacts table.

[5] As with MicroStrategy's motion for summary judgment of non-infringement, MicroStrategy presents some background to aid the Court in understanding the issues. This background is believed to be noncontroversial and is well-supported by the text of the patent itself (including the two provisional applications incorporated by reference), the expert declaration submitted by DataTern in its New York litigation, and the background discussion in the two Federal Circuit opinions on this patent.

A database can be described in terms of its structure, also known as a "schema." A database schema defines individual tables in the database, the data attributes in each table, and relationships among tables. An example of a relational database schema is depicted in Fig. 2 of the '502 patent (SOF ¶ 14):

FIG. 2



As represented by the schema depicted above from the '502 patent, the database includes a "TEmployee" table and a "TPerson" table. Since all employees are also people, the tables are related; the common attribute "id" allows one to look up the "TPerson" that corresponds to a particular "TEmployee." The line between the tables is annotated in the patent to indicate a 1:1 relationship between the tables. In other words, each entry in the TPerson table corresponds to exactly one entry in the TEmployee table, and vice-versa.

(b) Object-Oriented Programming

Object-oriented programming is a paradigm that uses "objects." At a high level, objects can perform tasks ("methods") and store data ("attributes"). For example, in an email program, an example of an object is a message. A second example of an object is a contact. An attribute of a message object is the message's subject; an attribute of a contact object is the person's name. An object's attribute can also be another object. For example, a second attribute of a message object is the sender, which may be represented as a contact object.

An "object model" represents all the classes that make up an object-oriented application and the relationships between those classes.[6] (SOF ¶ 15). The '502 patent, at Figure 3, depicts an example object model (SOF ¶ 16):

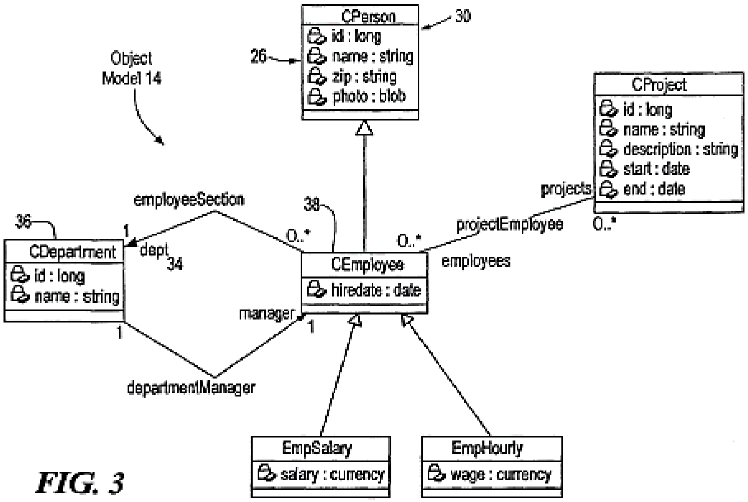


FIG. 3

The sample object model includes several classes: CPerson, CEmployee, EmpSalary (representing a salaried employee), EmpHourly (representing an hourly employee), CProject, and CDepartment. The CPerson class has four attributes: an "id," a "name," a "zip," and a "photo." Any object belonging to the CPerson class therefore has at least these four attributes. One such object might have the "name" attribute set to "Jane Doe"; another object might have the "name" attribute set to "John Smith." The CEmployee class is a subclass of CPerson, meaning it will have all the same attributes and behaviors, and an additional attribute "hiredate." This so-called "inheritance" relationship between CPerson and CEmployee is depicted in the patent's exemplary object model by an arrow.

2. Independent Claim 1

The '502 patent describes a method of mapping a database schema to an object model to

[6] A further description of the makeup of an "object model" is found in MicroStrategy's Motion for Summary Judgment of Non-infringement, filed contemporaneously herewith.

facilitate database access. (SOF ¶ 17.) In particular, a user selects an object model with classes that reflect the structure and data in the database. (SOF ¶ 18.) A mapping is then made between classes in the object model and database tables. (SOF ¶ 19.) Next, the map is used to create interface objects, which bridge the gap between data in the database and objects corresponding to classes in the object model. (SOF ¶ 20.) Finally, a runtime engine, through the interface objects, queries the database to obtain data. (SOF ¶ 21.) Independent claim 1 recites the following steps:

(1) selecting an object model;

(2) generating a map of at least some relationships between schema in the database and the selected object model;

(3) employing the map to create at least one interface object associated with an object corresponding to a class associated with the object oriented software application; and

(4) utilizing a runtime engine which invokes said at least one interface object with the object oriented application to access data from the relational database.

Each step is described in more detail in the following sections.

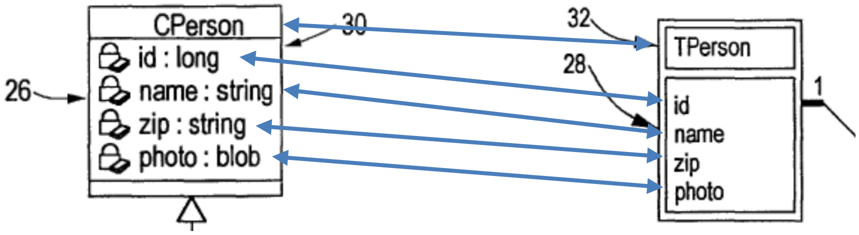
(a) "Selecting an object model"

The first step simply requires a user to pick an object model. The provisional patent, incorporated by reference into the '502 patent specification, confirms this step only requires a user to pick an existing object model from among choices. (SOF ¶ 22); Ex. D at 5 ("The designer tool allows a user to import an object model."); *id* at 29 ("You also need to import the object model by selecting a[n] . . . object model file"). In other words, a person satisfies the first step of the claim by mentally selecting the object model depicted in the patent's Figure 3 (shown above) from a set of preexisting choices.

(b) "Generating a map"

The second step, "generating a map" of specified relationships, involves a person identifying relationships between a database schema and the object model selected in the first step. (SOF ¶ 23.); '502 Patent, col. 2, lines 44-46 ("Mapping the object model to the relational database schema includes mapping a class attribute to a table column.”)); Ex. D (Provisional Application) at 5 (“To complete the mapping, a rich user interface is provided to enable the user to easily build a mapping between the object mode1 and database schema.”). This means the types of data and relationships embodied in the database are matched with corresponding classes and class attributes in the selected object model.

A person satisfies this step of the claim by correlating objects in the object model with tables in the schema. For example, this step is practiced when a person mentally associates a “Person” class in the object model with a “Person” table in the database, and identifies how the attributes in the Person class match up with columns in the Person table. (SOF ¶ 24); ’502 patent, col. 2, lines 53-58 (“the class attribute CPerson.name 26 maps to table column TPerson.name 28. . .[and] CPerson.id 30 maps to TPerson.id 32.”). This portion of the mapping is illustrated below:



(c) “Employing the map to create at least one interface object”

The third and fourth steps of the claim recite a known technique for accessing data from a relational database. In the third step, “interface objects” are created to access data from the database according to the mapping set forth by the user in step two’s mental process. (SOF ¶ 25.) According to the ’502 patent specification, an interface object is a front end to a database, providing an object-oriented programming application with database access. (SOF ¶ 26.) In DataTern’s New York action, it proposed that “interface object” simply means “an object by which the object oriented application accesses the relational database via a runtime engine.”

(SOF ¶ 13.) In other words, the “interface object” merely provides a means (via an “object”) to access a database and retrieve/store data consistent with the mapping from step two.

Creating “interface objects” based on a user’s mapping to access a database was well known long before the patent-in-suit was filed. For example, U.S. Patent No. 5,212,787 to Baker (Ex. E)[7] (cited during reexamination of the ’502 patent) discloses that a “requestor object located in the object-oriented environment sends a data access message requesting data located in a relational database . . . . This message is sent to an ***interface*** which creates a ***data object***. . . . The interface executes a data access routine and the results are placed into the data object,” which is returned to the requestor. (SOF ¶ 27) (*Id.* (emphasis added), Abstract; *see also id.* at 4:65-5:57.) Further, the “interface objects” in the Baker patent, like those recited in the ’502 claims, apply a mapping to specify what data in the database corresponds to which attributes in the data object. (SOF ¶ 28.) (*See id.* at 6:47-53: “a user may be mailing documents and desire to retrieve name and address, but not a telephone number. For each situation, a table in the Table Object 63 is used to pass parameters to the data access routine, directing the routine as to what data is to be placed in the data object and in what order.”)

(d) “Utilizing a runtime engine”

In the fourth step, a “runtime engine” reads and/or writes data in the database. According to the patent specification, the function of the runtime engine is to “access the relational

[7] It is appropriate to consider prior art in the context of deciding a § 101 motion. *See Mayo*, 132 S. Ct. at 1304 (“We recognize that, in evaluating the significance of additional steps, the § 101 patent-eligibility inquiry and, say, the § 102 novelty inquiry might sometimes overlap.”); *I/P Engine, Inc. v. AOL Inc.,* 576 Fed. Appx. 982, 994 fn.3 (Fed. Cir. 2014) (“There is, of course, some “overlap” between the eligibility analysis under section 101 and the obviousness inquiry under 35 U.S.C. § 103. … the section 101 inquiry is broader and more essential: it asks whether the claimed subject matter, stripped of any conventional elements, is “the kind of ‘discover[y]’ ” that the patent laws were intended to protect.”) (internal citations omitted).

database" using "the map . . . to drive its processing." (SOF ¶ 29.) In its New York case, DataTern proposed that "runtime engine" means "software, which is not directly part of the object-oriented application, that the object-oriented application uses to access the relational database." (SOF ¶ 11.) The New York court ultimately construed "runtime engine" to mean "software that (i) the object oriented software application depends on to run, (ii) must be running to execute the object oriented software application, (iii) uses the map in its processing, and (iv) is not part of the object oriented software application." (SOF ¶ 10.) Under either interpretation, a "runtime engine" is merely a separate piece of software used to access a database. The '502 patent confirms this, stating "[t]he interface objects and runtime engine perform read and write operations on the database." (SOF ¶30.) The '502 patent acknowledges that using software for database access was "well known" and there were multiple existing ways to do it. (SOF ¶ 31.)

3. Independent Claim 10

The only other independent claim in the '502 patent covers the nearly identical subject matter, but is written in "Beauregard" format and recites a "code generator." (SOF ¶ 32). In other words, it is a claim to an article of manufacture written as a computer-readable medium with instructions to implement a method like claim 1. Like claim 1, claim 10 includes requirements to "generate[] a map," "employ[] said map to create at least one interface object," and "invoke[] said interface object to access data from the relational database." (SOF ¶ 33.)

III. ARGUMENT

A. Legal Standards

1. Motion for Summary Judgment

Summary judgment under Rule 56(a) is appropriate where "the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to a judgment as a matter of law." Fed. R. Civ. P. 56(a). A material fact is one that "might affect the outcome of the suit under the governing law." *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248 (1986). "Factual disputes that are irrelevant or unnecessary will not be counted." *Id.* A genuine issue of material fact exists where the evidence with respect to the material fact in dispute "is such that a reasonable jury could return a verdict for the nonmoving party.*"* *Id.* Once the moving party has satisfied its burden, the burden shifts to the non-moving party to set forth specific facts showing that there is a genuine, triable issue. *Celotex Corp. v. Catrett*, 477 U.S. 317, 324 (1986).

Summary judgment is an appropriate vehicle to resolve a patent suit where the asserted patent is invalid for failure to satisfy 35 U.S.C. § 101. *See Planet Bingo, LLC v. VKGS LLC*, 576 Fed. Appx. 1005 (Fed. Cir. 2014) (affirming the district court's grant of summary judgment of invalidity under § 101); *see also Digitech Image Technologies, LLC v. Electronics for Imaging, Inc.*, 758 F.3d 1344 (Fed. Cir. 2014) (same).

2. Invalidity Under 35 U.S.C. § 101

Under 35 U.S.C. § 101, "[w]hoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." 35 U.S.C. § 101. The Supreme Court has articulated three exceptions to § 101's patent eligibility principles, finding the following categories of information are not patentable: (1) laws of nature, (2) physical phenomena, and (3) abstract ideas. *Alice*, 134 S. Ct. 2347, 2354 (2014).

Under *Alice*,[8] determining whether a claim covers an abstract idea and is thus unpatentable (or, alternatively, is a patent-eligible ***application*** of an idea) requires two steps. First, the Court determines "whether the claims at issue are directed to one of [the] patent

[8] The two-part test used in *Alice* was previously used by the Supreme Court in *Mayo*, 132 S. Ct. 1289, 1300 (2012) in the life-sciences context, but the *Alice* Court applied the test to patents—like DataTern's—that concern software.

eligible concepts," excluding abstract ideas (like mental processes). *Alice*, 134 S. Ct. at 2355. Second, the Court must ask "what else is there in the claims before us?" meaning "whether the additional elements transform the nature of the claim into a patent-eligible application." *Id.* (internal citations omitted).

Relevant to the first *Alice* step, mental processes are abstract ideas, which cannot be patented. Methods which can be performed mentally or that are "the equivalent of human mental work" are unpatentable abstract ideas, as they are "basic tools of scientific and technological work" that should be left open to all. *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1371 (Fed. Cir. 2011); *see Gottschalk v. Benson*, 409 U.S. 63, 67 (1972) (invalidating patent for converting binary-coded-decimal numbers to pure binary numbers because the entire process could be done mentally and "mental processes, and abstract intellectual concepts are not patentable."). Courts therefore "refuse[] to find processes patentable when they merely claimed a mental process standing alone and untied to another category of statutory subject matter[,] even when a practical application was claimed." *In re Comiskey*, 554 F.3d 967, 980 (Fed. Cir. 2009).

Under *Alice's* second step, appending conventional or insignificant post-solution activity is insufficient to transform an otherwise unpatentable abstract idea into a patentable application. *Alice*, 134 S. Ct. at 2359 (finding patent invalid where the "function performed by the computer at each step of the process is purely conventional.") (citations omitted); *see also Mayo Collaborative Services v. Prometheus Laboratories, Inc.*, 132 S. Ct. 1289, 1300 (2012) ("simply appending conventional steps, specified at a high level of generality, to laws of nature, natural phenomena, and abstract ideas cannot make those laws, phenomena, and ideas patentable"). "The notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance [as any] competent draftsman could attach some form of post-solution activity to almost any mathematical formula." *Parker v. Flook*, 437 U.S. 584, 590 (1978); *see also CyberSource*, 654 F.3d at 1371 (rejecting the notion that reciting a practical application for a calculation could alone make the invention patentable). Thus, where claim limitations add nothing significant beyond the mental process itself, those steps cannot transform the unpatentable mental process into patentable applications of that mental process. *See Mayo*, 132 S. Ct. at 1298.

B. A Determination that the '502 Patent Is Invalid Is Proper at this Stage

This Court previously deferred addressing Section 101 issues, citing the need to first resolve claim construction disputes because the patentability determination might "turn on the significance" of claim terms "interface object" and "runtime engine." Although this Court has not yet separately construed these terms, this motion is nonetheless ripe for decision because even under DataTern's proposed constructions the claims are unpatentable.

Following the Supreme Court's *Alice* decision, courts have routinely addressed (and granted) judgments of invalidity under § 101 prior to formal claim construction. *See buySAFE, Inc. v. Google, Inc.*, 765 F.3d 1350 (Fed. Cir. 2014) (affirming district court's grant of judgment on the pleadings where claims covered "creating familiar commercial arrangements by use of computers and networks"); *see Content Extraction and Transmission LLC v. Wells Fargo Bank, Nat'l Assoc. et al.*, No. 2013–1588, 2014 WL 7272219 (Fed. Cir. Dec. 23, 2014) (affirming district court's judgment of invalidity under § 101 decided prior to formal claim construction); *Ultramercial v. Hulu*, 772 F.3d 709, 713 (Fed. Cir. 2014) (affirming lower court's decision to grant defendant's "pre-answer motion to dismiss under Rule 12(b)(6) without formally construing the claims."). Notably, the *Ultramercial* decision included a concurring opinion where Judge Mayer recognized that "addressing section 101 at the outset of litigation will have a number of salutary effects."[9] *Id.*, 772 F.3d at 718–19.

To obviate any need to formally construe claims prior to a § 101 analysis, many courts have, for purposes of the analysis, construed the claims in a manner most favorable to patentee. *See Content Extraction*, 2014 WL 7272219 at *5 ("The district court construed the terms identified by CET in the manner most favorable to [CET]…."); *see also CMG Financial Services, Inc. v. Pacific Trust Bank, F.S.B.*, Case No. CV 11–10344, 2014 WL 4922349 (C.D. Cal. 2014) (granting summary judgment of invalidity under § 101 after "adopt[ing] Plaintiff's proposed claim constructions for purposes of this analysis."); *see also Gametek LLC v. Zynga, Inc.*, No. CV 13–2546, 2014 WL 1665090 at *3 (N.D. Cal. 2014) (same).

Like the above-cited cases, this Court may determine that the '502 patent claims are directed to unpatentable subject matter without formal claim construction, particularly given that—shortly after this Court's order on MicroStrategy's previous § 101 motion—the Southern District of New York completed claim construction proceedings for the patent. In New York, the terms "runtime engine" and "interface object" were either construed, or at least subject to proposed constructions by DataTern. (*See supra* § II.A). Thus, in accordance with Federal Circuit precedent, this Court may properly apply the New York court's construction,[10] or even DataTern's proposed constructions for "runtime engine" or "interface object" to assess

[9] The stated benefits were to (1) "conserve scarce judicial resources," (2) "provide[] a bulwark against vexatious infringement suits," and (3) "protecting the public" by proving a "tool for clearing the patent thicket, weeding out those patents that stifle innovation." *Id.* at 719.

[10] DataTern may dispute whether the New York court's construction of "runtime engine" is binding (ECF No. 103 at 3). MicroStrategy contends that it is. In any event, the result would be no different were the Court to alternatively adopt DataTern's proposed construction of "runtime engine" (rejected by the New York Court), for purposes of the § 101 analysis.

patentability. *See e.g., Content Extraction*, 2014 WL 7272219. As explained below, the claims fail to recite patent eligible subject matter regardless of which interpretation is applied.

C. Independent Claim 1 Is Invalid Under § 101

1. The Core Aspects of Claim 1 Are Drawn to an Unpatentable Mental Process

The ’502 patent fails the first step of the Supreme Court’s two-part *Alice* test for patentability because the core of the invention—mapping relationships between an object model and a database schema—represents an abstract idea that can be performed entirely by the human mind, and thus is not patentable. As shown below, each claim limitation is either a mental process or a conventional “post-solution” activity that is insufficient to confer patentability.

(a) “Selecting an object model”

The first step, “selecting an object model” merely involves the human mental process of choosing an object model from existing options. (*Supra* § II.B.2.(a)). Such a mental process falls outside the scope of patent eligible subject matter. *See CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1372 (Fed. Cir. 2011) (finding that limitation “‘obtaining information about other transactions’…can be performed by a human who simply reads records” and “such data-gathering steps cannot alone confer patentability”); *see also East Coast Sheet Metal Fabricating Corp. v. Autodesk, Inc.*, No. 12-cv-517-LM at 2-3. (D.N.H. Jan. 15, 2015) (finding that “obtaining a visual representation” was merely an abstract concept). DataTern’s previous opposition to MicroStrategy’s motion for judgment on the pleadings under § 101 did not purport to contest that this step could be satisfied by a mental process.

(b) “Generating a map”

Like the first step of claim 1, the second step, “generating a map” of relationships between the database schema and selected object model, can be performed on paper, by recording relationships between the database schema and object model. (*See supra*, § II.B.2.(b)). In fact, "[i]t would be difficult to conceive of a more abstract concept than "mapping." *East Coast Sheet Metal*, No. 12-cv-517-LM at 15 (where the claim recited mapping data describing a system to stored data); *CyberSource,* 654 F.3d at 1372-73 (finding a mapping claim step abstract because "a person may construct a map of credit card numbers. . .by writing down a list of credit card transactions made from a particular IP address") (internal quotes omitted). The Central District of California similarly determined that a process of "establishing relationships between document objects" was an abstract idea that could not be patented. *Bascom Research, LLC v. LinkedIn, Inc.*, No. 12–cv–06293, 2015 WL 149480 at *9 (N.D. Cal. 2015).

Thus, the subject matter of the '502 patent claims (limitations one and two) involve nothing more than mental processes that cannot satisfy step one of the *Alice* test.

2. The Remaining Steps Merely Add Insignificant Post-Solution Activity to the Unpatentable Mental Process and Do Not Transform the Claims into Patentable Subject Matter

The '502 patent likewise fails step two of the *Alice* test as the remaining claim limitations merely reflect post-solution conventional activity that does not transform the abstract idea into a patent eligible invention. Namely, once the core process of mapping relationships between the database schema and selected object model is complete, the remaining steps merely use the map to access data in the database using conventional methods.

(a) Creating an "interface object"

The third claim limitation, employing a map to create interface objects, merely reflects the conventional activity of creating an object to access a database in a manner consistent with the mapping generated in step two. The Court previously identified this term as one that may need construction to determine its significance. As explained above, using interface objects (even as DataTern interprets them) to access data in a database according to a user's mapping was well-known before the '502 patent was filed and reflects nothing more than the conventional use of existing computer technology. (*See supra,* § II. B.2.(c).);[11] *see also Planet Bingo, LLC v. VKGS LLC,* 576 Fed. Appx. 1005 (Fed. Cir. 2014) (finding patent invalid where the steps of storing and retrieving data using a computer were "purely conventional" and thus could not make the otherwise abstract idea patentable.)

(b) Using a "runtime engine"

The fourth claim limitation also merely applies conventional computing techniques to access data from the database and thus cannot save the claim. Similar to "interface objects" the Court stated the '502 patent's validity turned in part on the significance of the "runtime engine," thus the Court believed it was prudent to await that term's construction. (*See supra* § II.A)). Under DataTern's proposed construction, or the New York court's actual construction, a runtime engine merely reads and writes data to a database using conventional computer software techniques. (*See supra* § II.B.2.(d)).

Taken together or individually, the third and fourth steps simply limit the claim to a specific, but conventional, way for a computer to access a database. Since the Supreme Court's *Alice* decision, courts routinely find that applying conventional computer tasks to an abstract idea does not confer eligibility. For example, in *Planet Bingo*, the Federal Circuit determined the recitation of computer components throughout the claim, which performed steps to "select, store, and retrieve" information were insufficient for patentability where those steps "amount to a mere instruction to 'implemen[t]' an abstract idea 'on ... a computer.'" *Planet Bingo*, 576 Fed. Appx.

[11] In determining whether a step is "purely conventional or obvious," it is appropriate to look to evidence (such as prior art) that might also be relevant to a validity analysis under 35 U.S.C. §§ 102 and 103. *See Mayo*, 132 S. Ct. at 1304 ("We recognize that, in evaluating the significance of additional steps, the § 101 patent-eligibility inquiry and, say, the § 102 novelty inquiry might sometimes overlap.").

1005, 1008 (Fed. Cir. 2014) (quoting *Alice*, 134 S. Ct. at 2358). The patentee in *Planet Bingo* argued its patents covered more than an abstract idea because the invention included complex computer code with three distinct subparts. *Id.* The court invalidated the patents nonetheless finding "the function performed by the computer at each step of the process is purely conventional" (internal citations omitted). *Id.* at 1009.

Similarly in *East Coast Sheet Metal*, the court examined for patentability claims directed to computerized design of a ventilation system, which included mapping user-selected data to stored data about system components. No. 12-cv-517-LM at 2-3 (D.N.H. Jan. 15, 2015). The court first recognized that "it would be difficult to conceive of a more abstract concept than 'mapping.'" *Id.* at 15. The court then found that none of the computer-aided processes in the claims transformed the invention into patentable subject matter, stating:

> the patents-in-suit describe the inventions' computer programming as operating in the most generic of terms. They talk about data being processed, transferred, and stored using computer memory and a processor. There is no language in the patents-in-suit that describes the computer programming involved in the invention as operating in anything other than their "normal, expected manner…."

Id. at 23. The court thus held the patents invalid.

Finally, in *OpenText S.A. v. Box, Inc., et al.*, after determining that the patent's core concept—group collaboration and information sharing—was an abstract idea, the court turned to the question of whether the claims' additional computer-based elements could transform the claim into a patentable invention. No. 13-cv-04910 (N.D. Cal., Jan. 20, 2015). The court found that these additional components (a "server-based system," "web browser," "workgroup creator," and "security controls") were each "implementation-specific elements [that] were known prior to the invention" and thus the abstract idea remained unpatentable. *Id.* at 6.

Even DataTern itself has effectively conceded in the wake of the Supreme Court's *Alice* decision that its "interface object" and "runtime engine" limitations do not satisfy *Alice's* second step. Previously, DataTern argued to this Court that the "interface object" and "runtime engine" limitations "confer § 101 eligibility" just as the claimed "shadow credit record" and "shadow debit record" elements did for the *Alice* patents. DataTern cited to the then-precedential Federal Circuit panel decision in *Alice*. Since then, the Federal Circuit (*en banc*) and the Supreme Court have found that the "shadow [] record" limitations did not establish patentability, despite implicating computer functionality. *Alice*, 134 S. Ct. at 2359 (finding these steps to be "purely conventional…basic steps of a computer"); *CLS Bank Intern. v. Alice Corp. Pty. Ltd.*, 717 F.3d 1269, 1286-7 ("Nor does requiring the supervisory institution to create and adjust a 'shadow credit record' and a 'shadow debit record' narrow the claims from the realm of abstraction").

Like in *Alice*, the "interface object" and "runtime engine" limitations cannot transform the abstract ideas recited in DataTern's claims into a patentable invention.

D. Independent Claim 10 is Likewise Invalid Under § 101

Claim 10 covers subject matter nearly identical to claim 1, but written in Beauregard format, thus reciting a computer readable medium containing instructions for executing a method like that in claim 1.[12] Regardless of the statutory category under § 101 (process, machine, manufacture or composition of matter) a claim is drafted to invoke, courts "look to the underlying invention for patent-eligibility purposes." *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1374 (Fed. Cir. 2011) (invalidating a Beauregard claim drawn to an underlying unpatentable mental process). Like claim 1, claim 10 contains a limitation to "generate[] a map," "create[] at least one interface object," and use a "runtime engine that invokes [the] interface object." Each of these elements parallels an element in claim 1. In

[12] Claim 10 has one term that differs from claim 1, namely a "code generator" to create an interface object. Yet this limitation adds nothing of significance. A human programmer is a prototypical "code generator" and, as described above, generating interface objects is a well-known technique in the field of object-oriented programming. (*See supra*, § II.B.2.(c).)

addition, the Federal Circuit has consistently held that coupling an unpatentable mental process with a manufacture or machine does not render it patent-eligible. *See*, *e.g.*, *Parker v. Flook*, 437 U.S. 584, 590 (1978); *see also CyberSource*, 654 F.3d at 1371; *Prometheus*, 132 S. Ct. at 1300; *Fort Properties, Inc. v. American Master Lease LLC*, 671 F.3d 1317, 1323-4 (Fed. Cir. 2012). Accordingly, claim 10, although written in a different format than claim 1, is also invalid under § 101 because it covers an unpatentable mental process coupled with insignificant post-solution activity implemented using well-known techniques.

E. The Remaining Dependent Claims Are Likewise Invalid

Each of the remaining claims depends from claim 1 or claim 10, and continue to be unpatentable mental processes, or add nothing but trivial post-solution activity. For example, claims 2 and 11 (which relate to mapping a class attribute to a table column), claims 3-5 and 12-14 (which relate to mapping 1:1, 1:N and N:N relationships), and claim 41 (which relates to creating database schema) can each be performed with pen and paper, as shown in the illustrations in this brief. (*See supra*, § III.B.) Thus, like claims 1 and 10, these dependent claims are unpatentable mental processes under § 101. Other dependent claims add insignificant post-solution activity implemented using well-known techniques, and are thus likewise insufficient to confer patentability. (*See*, *e.g.*, claims 8 and 17 (creating multiple interface objects); claim 25 (using the runtime engine to execute a query of the database); and claim 28 (runtime engine includes a shared library).) Thus, the dependent claims are invalid under § 101.

IV. CONCLUSION

For the foregoing reasons, MicroStrategy respectfully requests that the Court grant its Motion for Summary Judgment of Invalidity for Unpatentable Subject Matter.

Dated: February 11, 2015

/s/ Adam J. Kessel
Adam Kessel (BBO #661,211)
Fish & Richardson, P.C.
One Marina Park Drive
Boston, MA 02110-1878
617-542-5070
617-542-8906 (fax)

Benjamin K. Thompson
(admitted *pro hac vice*)
bthompson@fr.com
Fish & Richardson, P.C.
1180 Peachtree St. NE
Atlanta, GA 30318
404-724-2844

Attorneys for Defendant
MICROSTRATEGY INCORPORATED.

CERTIFICATE OF SERVICE

I hereby certify that this document(s) filed through the ECF system will be sent electronically to the registered participants as identified on the Notice of Electronic Filing (NEF) and paper copies will be sent to those indicated as non-registered participants on this 11th of February, 2015.

/s/ Adam J. Kessel
Adam J. Kessel